CPE301 – SPRING 2019

Design Assignment 4B

Student Name: Mohamad Jundi

Student #: 8000321867

Student Email: jundi@unlv.nevada.edu

Primary Github address: https://github.com/MohamedJundi1994/Submission\_DA.git

Directory: Documents\School\CPE 301\Repository\CPE\_301\DesignAssignments\DA4B

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

Using Breadboard and lab power supply for motors,

USB port => Xplained Mini => Potentiometer => Stepper

USB port => Xplained Mini => Potentiometer => Servo

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

My Code (Stepper):

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

int adc\_value; // variable used to hold ADC value

int main(void)

{

DDRB = 0xFF; // Used to set all pins as output

PORTB = 0; // Used to make initial value 0

ADMUX = (0 << REFS1) | (1 << REFS0); // AREF, internal Vref turned off

ADCSRA = 0b11100111; // ADC enabled, ADC will start conversion, ADC auto trigger enabled, division factor set to 128

TCCR1A = (1 << COM1A1); // Used to clear OC1A on compare match

TCCR1B = (1 << WGM12) | (1 << CS12) | (1 << CS10); // Used for setting mode to CTC and setting a prescaler of 1024

adc\_value = ADC;

OCR1A = adc\_value; // OCR1A holds adc value

while(1)

{ // Following code used to rotate stepper motor counter clockwisse with full step sequence

if ((OCR1A >= 0) && (OCR1A < 100))

{

PORTB = 0x09;

*\_delay\_ms*(5);

PORTB = 0x03;

*\_delay\_ms*(5);

PORTB = 0x06;

*\_delay\_ms*(5);

PORTB = 0x0c;

*\_delay\_ms*(5);

OCR1A = adc\_value;

}

if ((OCR1A >= 100) && (OCR1A < 200))

{

PORTB = 0x09;

*\_delay\_ms*(50);

PORTB = 0x03;

*\_delay\_ms*(50);

PORTB = 0x06;

*\_delay\_ms*(50);

PORTB = 0x0c;

*\_delay\_ms*(50);

OCR1A = adc\_value;

}

if ((OCR1A >= 200) && (OCR1A < 300))

{

PORTB = 0x09;

*\_delay\_ms*(100);

PORTB = 0x03;

*\_delay\_ms*(100);

PORTB = 0x06;

*\_delay\_ms*(100);

PORTB = 0x0c;

*\_delay\_ms*(1000);

OCR1A = adc\_value;

}

}

}

My Code (Servo):

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

int adc\_value; // variable used to hold ADC value

int main(void)

{

DDRB = 0xFF; // Used to set all pins as output for B

DDRD = 0xFF; // Used to set all pins as output for D

ADMUX = (1 << REFS1) | (1 << REFS0); // Internal 1.1 voltage reference

ADCSRA = 0b10100110; // ADC enabled, ADC auto trigger enabled

TCCR1A = (1 << COM1A1) | (1 << COM1B1) | (1 << WGM11) ; // Used to clear OC1A on compare match fast PWM

TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS12) | (1 << CS10); // Used for setting a prescaler of 1024 and Fast PWM

ICR1 = 5000; // ICR1 value set

while(1)

{

while((ADCSRA & (1 << ADIF)) == 0); // while conversion starts

adc\_value = ADC;

if(adc\_value == 0) // If pot value is at it's lowest enter if statement

{

OCR1A = 0; // Used to turn to 0

*\_delay\_ms*(600); // Give time for transition

}

else if(adc\_value == 255) // If pot value is at it's highest enter if statement

{

OCR1A = 535; // Used to turn to 180

*\_delay\_ms*(600); // Give time for transition

}

}

}

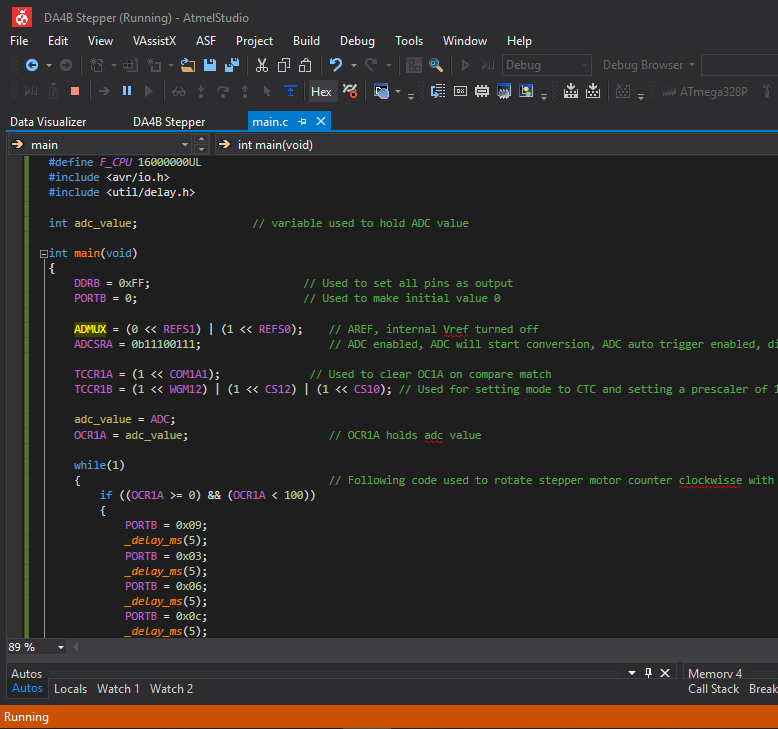
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

All code is in number 2

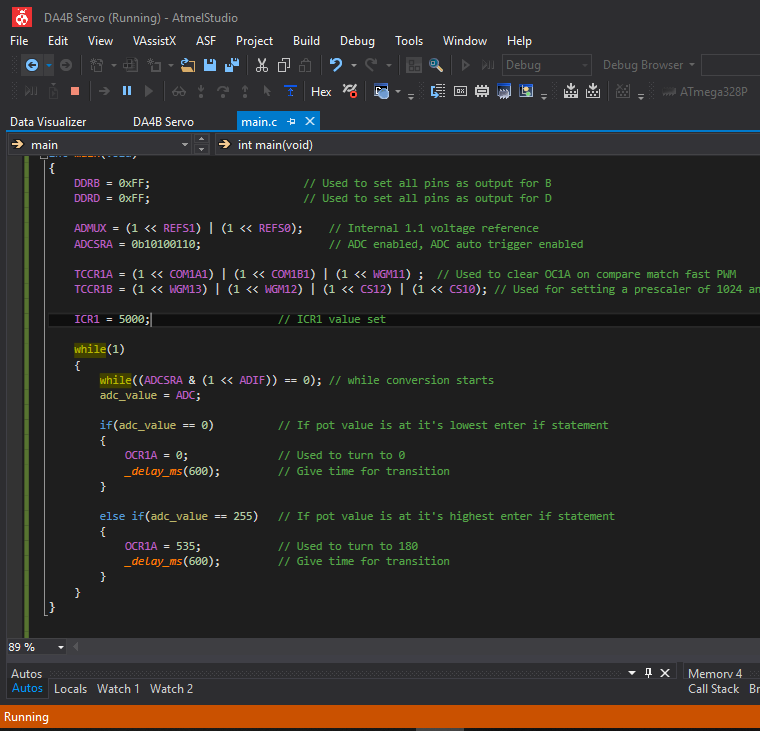
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

**NEXT PAGE =>**

Stepper:

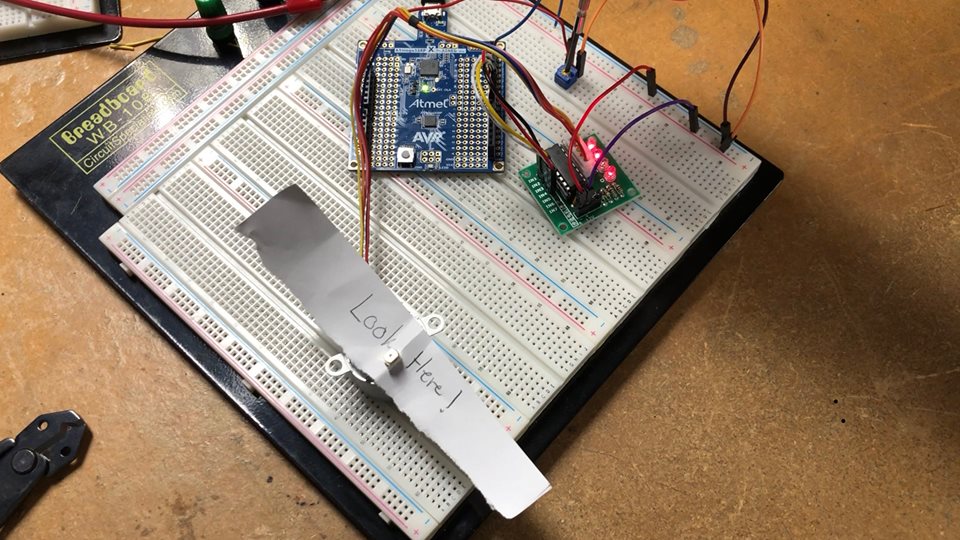


Servo:

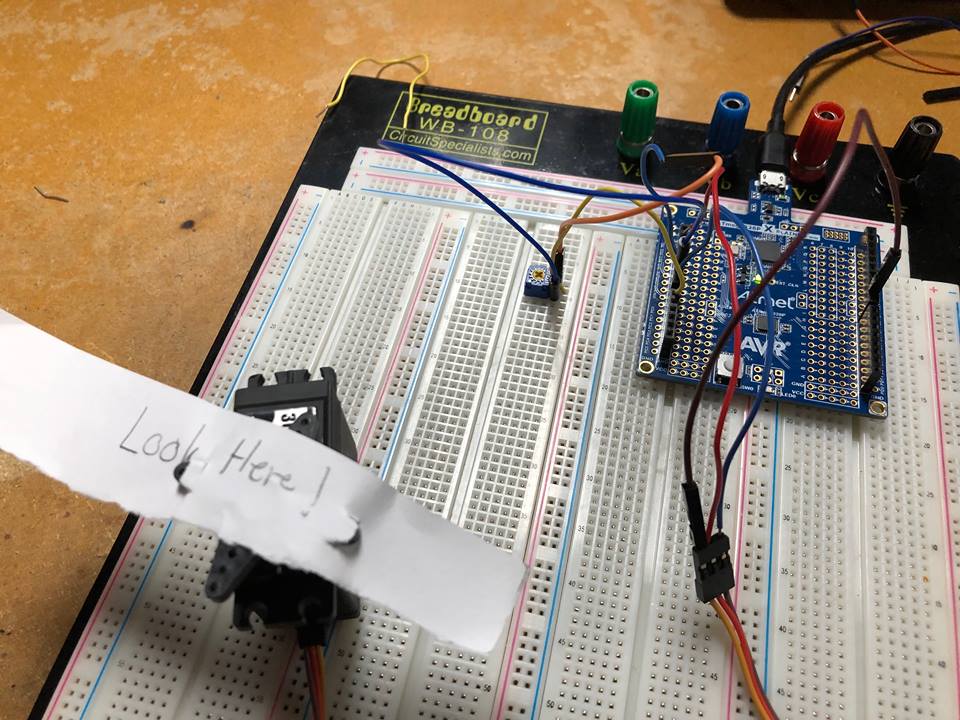


1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

Stepper:



Servo:



1. **VIDEO LINKS OF EACH DEMO**

Stepper link:

<https://www.youtube.com/watch?v=iswg6FjdLQQ&feature=share&fbclid=IwAR04cIo9heSQUlZ_Ll_ZWY4C-uRyDH0MNMLBfF2rHtdEc8fVtKVvdWl95ys>

Servo link:

<https://www.youtube.com/watch?v=wMj9lPIKkFo&feature=share&fbclid=IwAR2vUs-psmK6a__Kdzrx6QC60SKX0_iip8OfiCfN2-Ixu-9Xeoy3AN17Ing>

1. **GITHUB LINK OF THIS DA**

Link: https://github.com/MohamedJundi1994/Submission\_DA.git

This assignment submission is my own, original work.

MOHAMAD JUNDI